

IN THE CLAIMS:

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1. - 64. (Cancelled).

65. (Previously Presented) A computer-implemented method for automatically generating a new graphical program, comprising:

executing a graphical program generation (GPG) program;

the GPG program receiving information, wherein the information specifies functionality of the new graphical program, wherein the information does not specify specific objects for the new graphical program; and

the GPG program automatically generating the new graphical program in response to said information specifying the functionality of the new graphical program, wherein the new graphical program implements the specified functionality, and wherein the new graphical program comprises a plurality of interconnected nodes that visually indicate the functionality of the new graphical program;

wherein said automatically generating the new graphical program is performed without direct user input specifying the new graphical program.

66. (Previously Presented) The method of claim 65, wherein said automatically generating the new graphical program is performed without direct user input selecting the plurality of nodes and without direct user input specifying the interconnections between the plurality of nodes.

67. (Previously Presented) The method of claim 65, wherein the new graphical program comprises a block diagram portion comprising the plurality of interconnected nodes and a user interface portion; and

wherein said automatically generating the new graphical program includes automatically generating the block diagram portion and the user interface portion.

68. (Previously Presented) The method of claim 65, wherein said automatically generating the new graphical program comprises:

creating a plurality of graphical program objects in the new graphical program;
and

interconnecting the plurality of graphical program objects in the new graphical program;

wherein the interconnected plurality of graphical program objects comprise at least a portion of the new graphical program.

69. (Previously Presented) The method of claim 65, wherein said automatically generating the new graphical program comprises:

creating one or more user interface objects in the new graphical program, wherein the one or more user interface objects perform one or more of providing input to or displaying output from the new graphical program.

70. (Previously Presented) The method of claim 65,

wherein the new graphical program is a virtual instrument.

71. (Previously Presented) The method of claim 65,

wherein the GPG program is a graphical program.

72. (Previously Presented) The method of claim 65,

wherein the information received by the GPG program specifies a computational process; and

wherein the GPG program is operable to generate a new graphical program that implements the specified computational process.

73. (Previously Presented) The method of claim 65,

wherein the information received by the GPG program specifies an algorithm; and
wherein the GPG program is operable to generate a new graphical program that implements the specified algorithm.

74. (Previously Presented) The method of claim 65,

wherein the information received by the GPG program specifies a state diagram;
and

wherein the GPG program is operable to generate a new graphical program that implements the specified state diagram.

75. (Previously Presented) The method of claim 65,

wherein the information received by the GPG program specifies a prototype; and
wherein the GPG program is operable to generate a new graphical program that implements the specified prototype.

76. (Previously Presented) The method of claim 65,

wherein the information received by the GPG program specifies a test executive sequence; and

wherein the GPG program is operable to generate a new graphical program that implements the specified test executive sequence.

77. (Previously Presented) The method of claim 65,

wherein said GPG program receiving information comprises the GPG program receiving user input specifying desired functionality of the new graphical program; and

wherein the GPG program is operable to generate a new graphical program that implements the specified desired functionality.

78. (Previously Presented) The method of claim 77,

wherein the GPG program comprises a graphical programming development environment application.

79. (Previously Presented) The method of claim 65,
wherein the information received by the GPG program specifies an instrumentation function; and
wherein the GPG program is operable to generate a new graphical program that implements the specified instrumentation function.
80. (Previously Presented) The method of claim 79,
wherein the instrumentation function comprises one or more of:
a test and measurement function; or
an industrial automation function.
81. (Previously Presented) The method of claim 65,
wherein the information received by the GPG program comprises information regarding an existing program having program functionality; and
wherein the GPG program is operable to generate a new graphical program that implements at least a portion of the program functionality of the existing program.
82. (Previously Presented) The method of claim 81,
wherein the existing program is a graphical program.
83. (Previously Presented) The method of claim 65,
wherein the GPG program is operable to generate a plurality of new graphical programs, depending on the received information.
84. (Previously Presented) The method of claim 65,
wherein the new graphical program generated by the GPG program has program functionality; and
wherein the GPG program is operable to determine at least a portion of the program functionality independently of the received information.

85. (Previously Presented) The method of claim 65,

wherein the GPG program is operable to generate the new graphical program such that the new graphical program implements additional functionality in addition to the functionality specified by the received information.

86. (Previously Presented) The method of claim 65,

wherein the new graphical program comprises graphical program code; and
wherein the GPG program is operable to receive code generation information specifying how to generate at least a portion of the graphical program code.

87. (Previously Presented) The method of claim 65,

wherein said GPG program automatically generating the new graphical program comprises the GPG program calling an application programming interface (API) enabling the automatic generation of a graphical program.

88. (Previously Presented) The method of claim 65,

wherein said GPG program automatically generating the new graphical program comprises the GPG program automatically requesting a server program to generate the new graphical program.

89. (Previously Presented) The method of claim 88,

wherein the server program is an application instance of a graphical programming environment.

90. (Previously Presented) The method of claim 65,

wherein the GPG program comprises a client portion and a server portion; and
wherein the client portion is operable to utilize an application programming interface (API) in order to direct the server program to automatically generate the new graphical program.

91. (Previously Presented) The method of claim 90,

wherein the client portion of the GPG program executes in a first computer system;

wherein the server portion of the GPG program executes in a second computer system; and

wherein the first computer system is connected to the second computer system.

92. (Previously Presented) The method of claim 65, further comprising:

executing the new graphical program;

wherein the new graphical program is operable to perform the specified functionality during execution.

93. (Previously Presented) The method of claim 65,

wherein the new graphical program implements only a portion of the specified functionality.

94. (Previously Presented) The method of claim 65, wherein the new graphical program is a partial program, the method further comprising:

adding additional graphical code to the new graphical program, in response to manual user input, in order to complete the new graphical program.

95. (Previously Presented) The method of claim 65,

wherein said automatically generating the new graphical program comprises including at least one graphical program object in the new graphical program.

96. (Previously Presented) The method of claim 95, wherein the new graphical program includes a block diagram, wherein the at least one graphical program object comprises a function node placed in the block diagram.

97. (Previously Presented) The method of claim 95, wherein the new graphical program includes a block diagram, wherein the at least one graphical program object comprises a automatic structure placed in the block diagram.

98. (Previously Presented) The method of claim 95, wherein the new graphical program includes a user interface panel, wherein the at least one graphical program object comprises a user interface object placed in the user interface panel.

99. (Previously Presented) The method of claim 98, wherein the user interface object is a user interface input object placed in the user interface panel for performing one or more of: viewing input to the new graphical program; or providing input to the new graphical program.

100. (Previously Presented) The method of claim 98, wherein the user interface object is a user interface output object placed in the user interface panel for viewing output of the new graphical program.

101. (Previously Presented) The method of claim 98, wherein the new graphical program also includes a block diagram, wherein the user interface object is a user interface input object placed in the user interface panel for performing one or more of: viewing input to the block diagram; or providing input to the new graphical program.

102. (Previously Presented) The method of claim 98, wherein the new graphical program also includes a block diagram, wherein the user interface object is a user interface output object placed in the user interface panel for viewing output from the block diagram.

103. (Previously Presented) The method of claim 65, wherein said automatically generating the new graphical program comprises:

including a first graphical program object and a second graphical program object in the new graphical program; and

connecting the first graphical program object to the second graphical program object.

104. (Previously Presented) The method of claim 103, wherein said connecting the first graphical program object to the second graphical program object comprises connecting an input of the first graphical program object to an output of the second graphical program object.

105. (Previously Presented) The method of claim 65,
wherein the GPG program is a graphical program;
wherein the GPG program includes at least one object creation node for automatically creating at least one graphical program object in the new graphical program; and
wherein said generating the new graphical program comprises including the at least one graphical program object in the new graphical program.

106. (Previously Presented) The method of claim 105, wherein the GPG program further includes a property node, the method further comprising:
the property node getting or setting a property of the graphical program object in response to said executing the GPG program.

107. (Previously Presented) The method of claim 106, wherein the object creation node outputs a reference to the graphical program object;
wherein the property node receives the reference as input to the graphical program object; and
wherein the property node gets or sets a property of the graphical program object specified by the reference to the graphical program object.

108. (Previously Presented) The method of claim 105, wherein the GPG program further includes an invoke node; the method further comprising:
the invoke node invoking a method on the graphical program object in response to said executing the GPG program.

109. (Previously Presented) The method of claim 108, wherein the object creation node outputs a reference to the graphical program object;

wherein the invoke node receives as input the reference to the graphical program object; and

wherein the invoke node invokes a method on the graphical program object specified by the reference to the graphical program object.

110. (Previously Presented) The method of claim 105, further comprising:

configuring the object creation node of the GPG program;

wherein said configuring comprises specifying a graphical program object class for the object creation node; and

wherein the at least one graphical program object included in the new graphical program is of the graphical program object class.

111. (Previously Presented) The method of claim 65,

wherein the GPG program is a graphical program; and

wherein the GPG program includes a graphical program creation node for automatically creating the new graphical program.

112. (Previously Presented) A computer-implemented method for automatically modifying an existing graphical program, comprising:

executing a GPG program;

the GPG program receiving information during program execution, wherein the information specifies functionality to add to the existing graphical program, wherein the information does not specify specific objects to add to the existing graphical program;

the GPG program automatically modifying the existing graphical program in order to implement the specified functionality, in response to receiving the information, wherein the existing graphical program comprises a plurality of interconnected nodes that visually indicate the specified functionality of the existing graphical program;

wherein said automatically modifying the existing graphical program modifies the existing graphical program without direct user input specifying the modification to the existing graphical program during said modifying.

113. (Previously Presented) The method of claim 112, wherein said automatically modifying the existing graphical program modifies the existing graphical program without direct user input specifying modification to the plurality of nodes and without direct user input specifying modification to the interconnections between the plurality of nodes.

114. (Previously Presented) The method of claim 112,

wherein said modifying the existing graphical program comprises adding graphical code to the existing graphical program.

115. (Previously Presented) The method of claim 112,

wherein said receiving information during program execution comprises receiving user input specifying desired functionality to add to the existing graphical program.

116. (Previously Presented) A computer-implemented method for automatically generating a new graphical program, comprising:

providing information specifying functionality of the new graphical program, wherein the information does not specify specific objects for the new graphical program;

executing a graphical program generation (GPG) program;

the GPG program automatically generating the new graphical program using said information, wherein the new graphical program implements the specified functionality, and wherein the new graphical program comprises a plurality of interconnected nodes that visually indicate the functionality of the new graphical program;

wherein said automatically generating the new graphical program creates the new graphical program without direct user input specifying the new graphical program.

117. (Previously Presented) The method of claim 116, wherein said automatically generating the new graphical program creates the new graphical program without direct user input selecting the plurality of nodes and without direct user input specifying the interconnections between the nodes.

118. (Previously Presented) A memory medium for automatically generating a new graphical program, the memory medium comprising program instructions executable to:

receive information, wherein the information specifies functionality of the new graphical program, wherein the information does not specify specific objects for the new graphical program;

automatically generate the new graphical program in response to said information specifying the functionality of the new graphical program, wherein the new graphical program implements the specified functionality, wherein the new graphical program comprises a plurality of interconnected nodes that visually indicate the functionality of the new graphical program, and wherein said automatically generating the new graphical program creates the new graphical program without direct user input specifying the new graphical program.

119. (Previously Presented) The memory medium of claim 118, wherein said automatically generating the new graphical program creates the new graphical program without direct user input selecting the plurality of nodes and without direct user input specifying the interconnections between the nodes.

120. (Previously Presented) The memory medium of claim 118, wherein the new graphical program comprises a block diagram portion comprising the plurality of interconnected nodes and a user interface portion;

wherein said automatically generating the new graphical program includes generating the block diagram portion and the user interface portion.

121. (Previously Presented) A system for automatically generating a new graphical program, the system comprising:

a processor coupled to a memory, wherein the memory stores a graphical program generation (GPG) program;

wherein the processor is operable to execute the GPG program in order to:

receive information specifying functionality of the new graphical program, wherein the information does not specify specific objects for the new graphical program; and

automatically generate the new graphical program in response to said information specifying the functionality of the new graphical program, wherein the new graphical program implements the specified functionality, and wherein the new graphical program comprises a plurality of interconnected nodes that visually indicate the functionality of the new graphical program, and wherein the new graphical program is automatically generated without direct user input specifying the new graphical program.

122. (Previously Presented) The system of claim 121, wherein said automatically generating the new graphical program creates the new graphical program without direct user input selecting the plurality of nodes and without direct user input specifying the interconnections between the nodes.

123. (Previously Presented) The system of claim 121, wherein the new graphical program comprises a block diagram portion comprising the plurality of interconnected nodes and a user interface portion;

wherein said automatically generating the new graphical program includes generating the block diagram portion and the user interface portion.

124. (Previously Presented) The system of claim 121, wherein said automatically generating the new graphical program comprises:

creating a plurality of graphical program objects in the new graphical program without direct user input specifying the plurality of graphical program objects; and

interconnecting the plurality of graphical program objects in the new graphical program without direct user input specifying the interconnections between the nodes; wherein the interconnected plurality of graphical program objects comprise at least a portion of the new graphical program.

125. (Previously Presented) The memory medium of claim 118, wherein, in automatically generating the new graphical program, the program instructions are executable to:

automatically create a plurality of graphical program objects in the new graphical program; and

automatically interconnect the plurality of graphical program objects in the new graphical program;

wherein the interconnected plurality of graphical program objects comprise at least a portion of the new graphical program.

126. (Previously Presented) The memory medium of claim 118, wherein, in automatically generating the new graphical program, the program instructions are executable to:

automatically create one or more user interface objects in the new graphical program, wherein the one or more user interface objects perform one or more of providing input to or displaying output from the new graphical program.

127. (Previously Presented) The memory medium of claim 118,

wherein the new graphical program implements a measurement function.

128. (Previously Presented) The memory medium of claim 118, wherein, in receiving information, the program instructions are executable to:

receive user input specifying desired functionality of the new graphical program, wherein the generated new graphical program implements the specified desired functionality.

129. (Previously Presented) The memory medium of claim 118,
wherein said receiving and said generating are performed by a graphical programming development environment application.
130. (Previously Presented) The memory medium of claim 118,
wherein the received information comprises information regarding an existing program having program functionality; and
wherein the generated new graphical program implements at least a portion of the program functionality of the existing program.
131. (Previously Presented) The memory medium of claim 118, wherein the program instructions are further executable to:
automatically generate a plurality of new graphical programs, depending on the received information.
132. (Previously Presented) The memory medium of claim 118,
wherein the generated new graphical program implements additional functionality in addition to the functionality specified by the received information.
133. (Previously Presented) The memory medium of claim 118,
wherein the new graphical program implements only a portion of the specified functionality.
134. (Previously Presented) The memory medium of claim 118, wherein the new graphical program is a partial program, and wherein the program instructions are further executable to:
add additional graphical code to the new graphical program in response to manual user input in order to complete the new graphical program.

135. (Previously Presented) A computer-implemented method for automatically generating a new graphical program, comprising:

receiving information, wherein the information specifies functionality of the new graphical program, wherein the information does not specify specific objects for the new graphical program; and

automatically generating the new graphical program in response to said information specifying the functionality of the new graphical program, wherein the new graphical program implements the specified functionality, and wherein the new graphical program comprises a plurality of interconnected nodes that visually indicate the functionality of the new graphical program;

wherein said automatically generating the new graphical program is performed without direct user input including the plurality of nodes or connecting the plurality of nodes.

136. (Previously Presented) The method of claim 135, wherein said automatically generating the new graphical program comprises:

automatically creating a plurality of graphical program objects in the new graphical program; and

automatically interconnecting the plurality of graphical program objects in the new graphical program;

wherein the interconnected plurality of graphical program objects comprise at least a portion of the new graphical program.

137. (Previously Presented) The method of claim 135, wherein said automatically generating the new graphical program comprises:

automatically creating one or more user interface objects in the new graphical program, wherein the one or more user interface objects perform one or more of providing input to or displaying output from the new graphical program.

138. (Previously Presented) The method of claim 135,

wherein the new graphical program implements a measurement function.

139. (Previously Presented) The method of claim 135,

wherein said receiving information comprises receiving user input specifying desired functionality of the new graphical program; and

wherein the generated new graphical program implements the specified desired functionality.

140. (Previously Presented) The method of claim 135,

wherein said receiving and said generating are performed by a graphical programming development environment application.

141. (Previously Presented) The method of claim 135,

wherein the received information comprises information regarding an existing program having program functionality; and

wherein the generated new graphical program implements at least a portion of the program functionality of the existing program.

142. (Previously Presented) The method of claim 135, further comprising:

automatically generating a plurality of new graphical programs, depending on the received information.

143. (Previously Presented) The method of claim 135,

wherein the generated new graphical program implements additional functionality in addition to the functionality specified by the received information.

144. (Previously Presented) The method of claim 135,

wherein the new graphical program implements only a portion of the specified functionality.

145. (Previously Presented) The method of claim 135, wherein the new graphical program is a partial program, the method further comprising:

adding additional graphical code to the new graphical program, in response to manual user input, in order to complete the new graphical program.